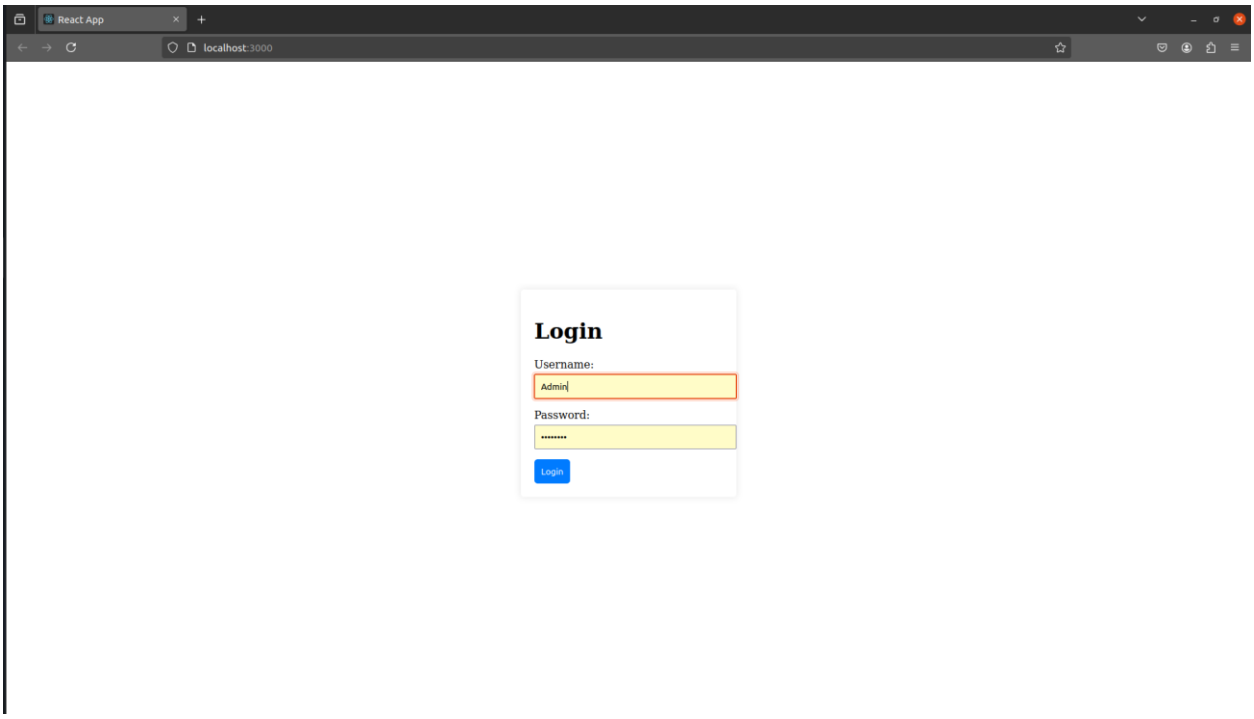
Dashboard Development Assignment Documentation

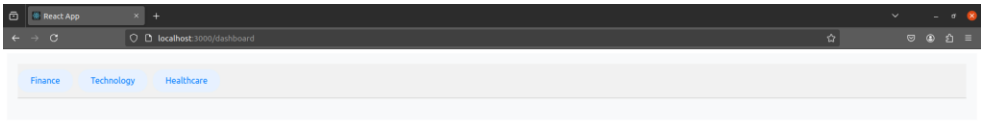
Git repository: https://github.com/ShwetaPokale/dashboard

Frontend:

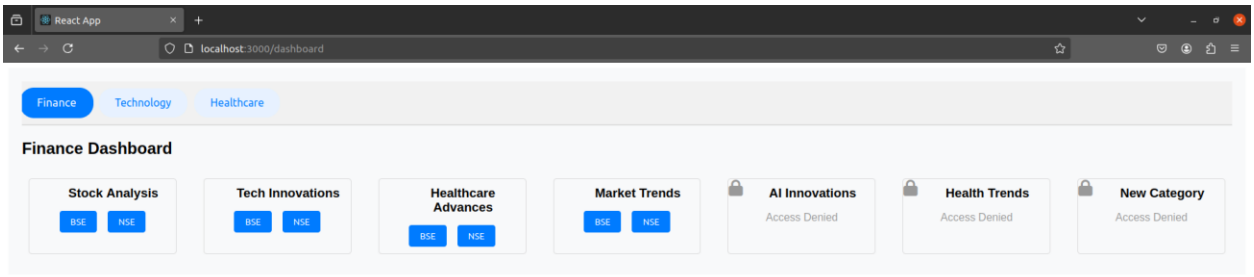
Login:



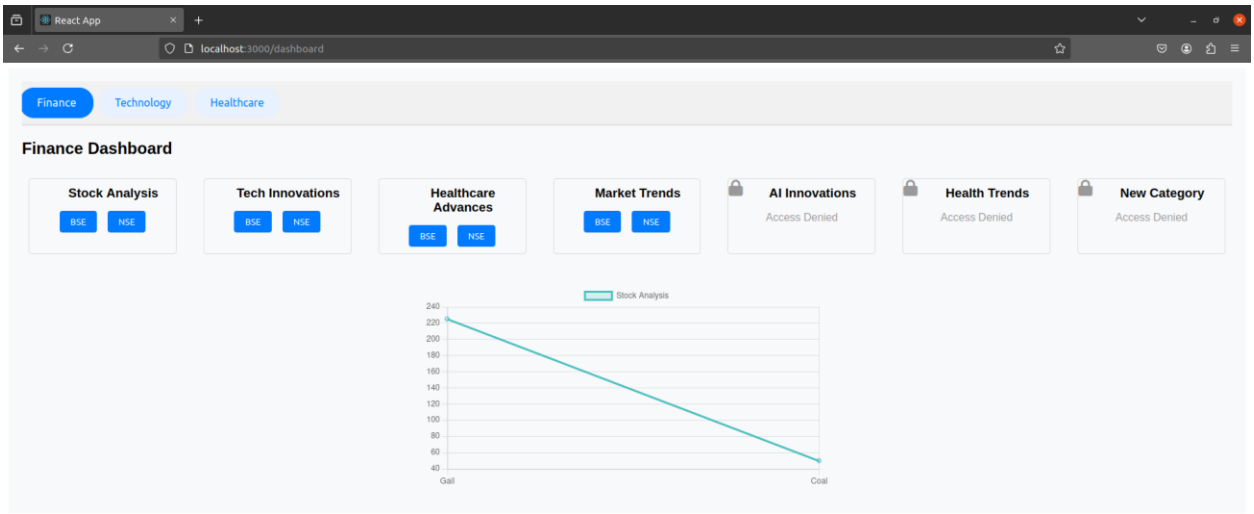
Dashboard
Sections:



Category Page:



Charts:



Backend:

Architecture

- **Framework**: NestJS (Node.js version 20.14, NestJS version 10.4.4)
- **Database**: PostgreSQL
- **ORM**: TypeORM for database management and object-relational mapping.

Database Schema

Tables:

- **users**: Stores user information and credentials.
- **bse_data_source**: Contains data related to the Bombay Stock Exchange (BSE).
- **categories**: Defines various categories for data organization.
- **category_charts**: Associates categories with charts for visualization.
- **charts**: Stores information about charts used in the dashboard.
- **data_sources**: Represents different data sources available in the application.
- **nse_data_source**: Contains data related to the National Stock Exchange (NSE).
- **sections**: Represents different sections in the dashboard.
- **user_categories**: Manages user-specific access to categories.

Backend Setup and API Usage

1. Install Dependencies:
   Path: dashboard-backend
   command: npm install

2. **Database Configuration**: Ensure PostgreSQL is configured correctly. Update your database credentials in the configuration file.
   Path: dashboard-backend/src/config.ts

3. **Start the Application**: Run the backend server using:
   Path: dashboard-backend
   command: npm start / npm run start

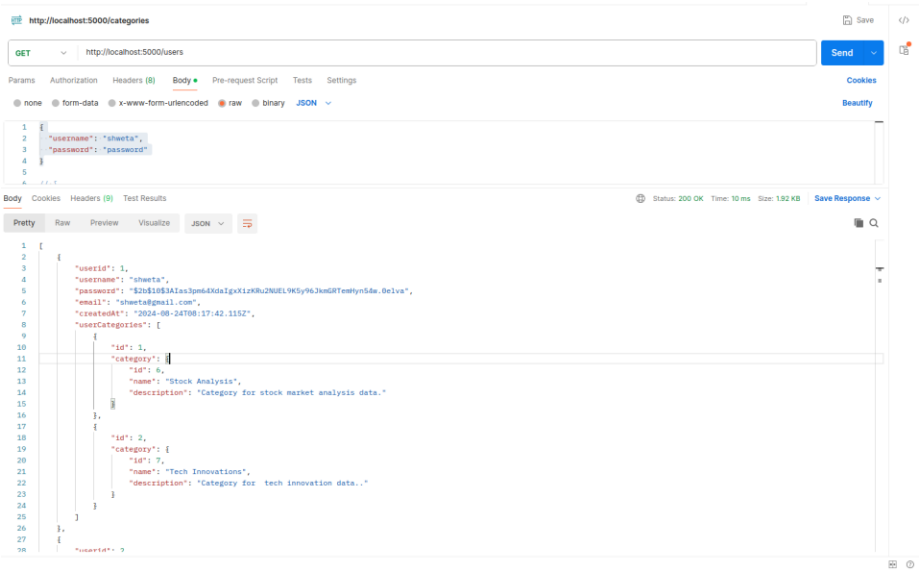
4. API Endpoints:

Users

- **Create User Endpoint**: POST /users
- While creating user can give access to categories based on category id.
- **Request Body**:

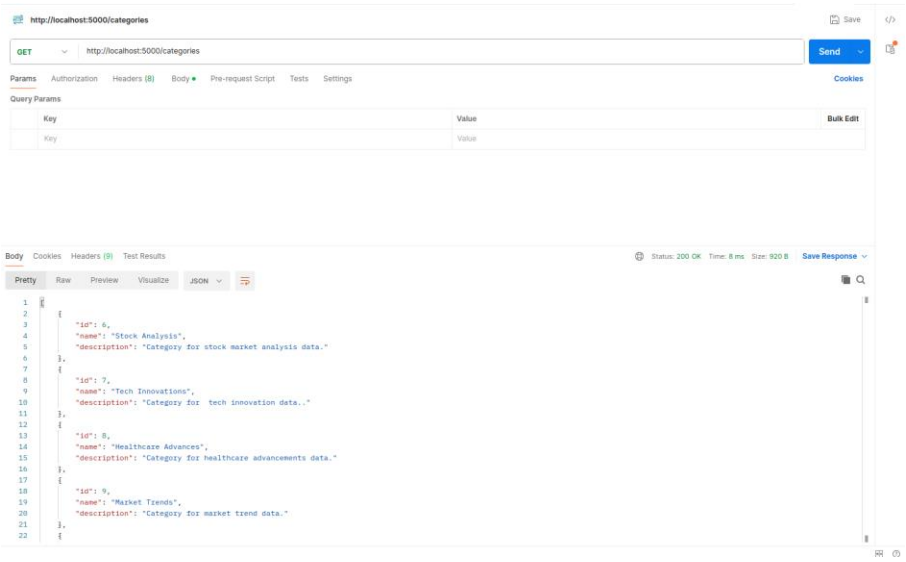
```
{
  "username": "string",
  "password": "string",
  "email": "string",
  "categoryIds": [number]
}
```

- **Get All Users Endpoint**: GET /users
- **Get User by ID Endpoint**: GET /users/:id
- **Update User Endpoint**: PUT /users/:id
- **Delete User Endpoint**: DELETE /users/:id



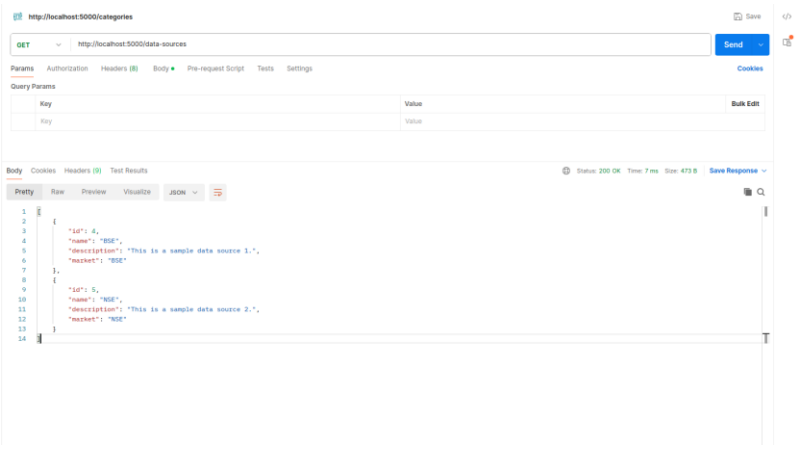
Categories:

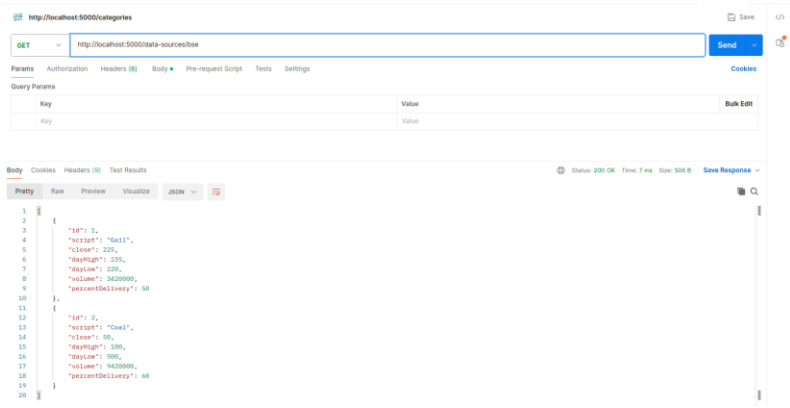
- **Get All Categories**: GET /categories
- **Get Categories** by User and Section: GET /categories/user/:userId/section/:sectionId
- **Get Category by ID**: GET /categories/:id
- **Create Category**: POST /categories
- **Update Category**: PUT /categories/:id
- **Delete Category**: DELETE /categories/:id

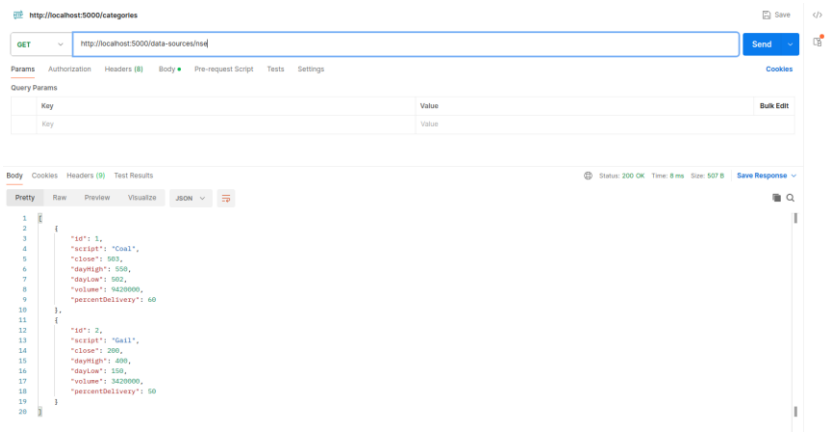


Data Sources:

- **Create Data Source Endpoint**: POST /data-sources
- **Get All Data Sources**
- **Endpoint**: GET /data-sources
- **Get Market Data Endpoint**: GET /data-sources/:type
    - **Response**: Market data (BSE or NSE).
- **Create Market Data Endpoint**: POST /data-sources/:type
- **Update Data Source Endpoint**: PUT /data-sources/:id
- **Delete Data Source Endpoint**: DELETE /data-sources/:id

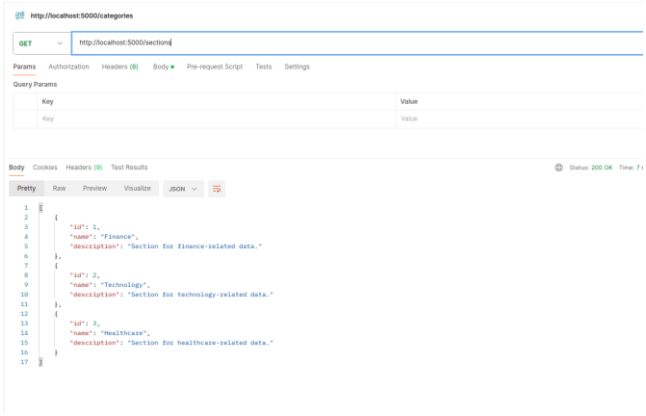
- 





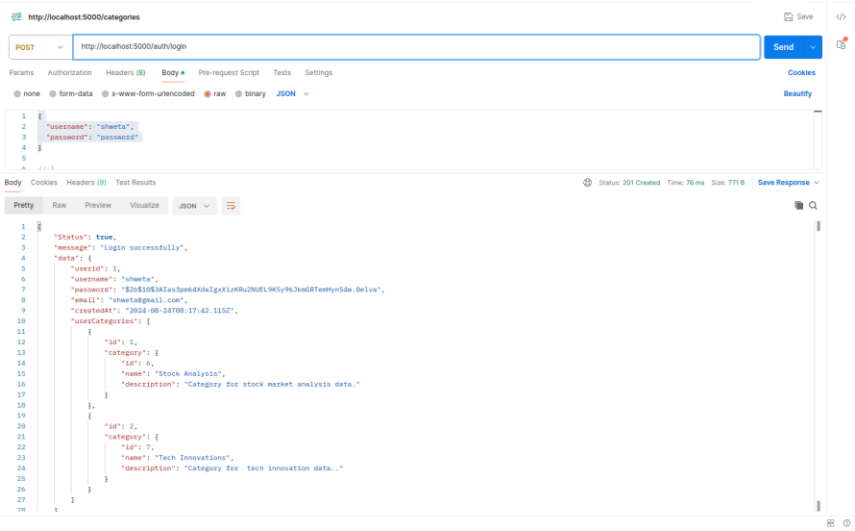
Sections:

- **Get All Sections**: GET /sections
- **Get Section by ID**: GET /sections/:id
- **Create Section**: POST /sections
- **Update Section**: PUT /sections/:id
- **Delete Section**: DELETE /sections/:id



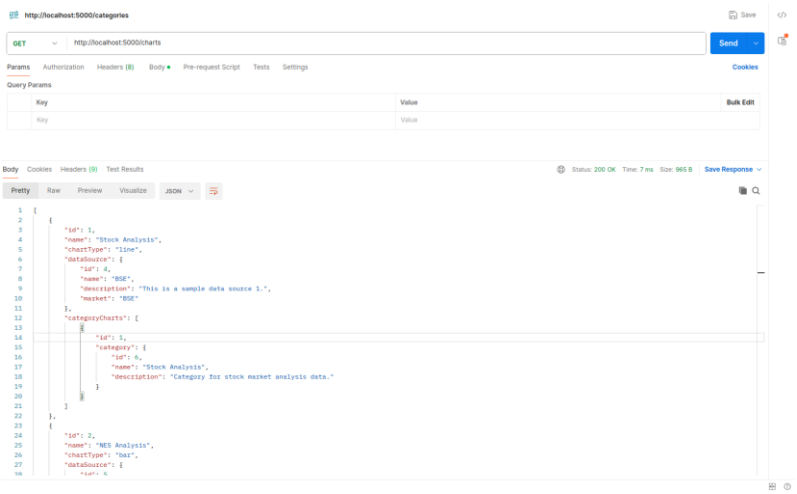
Login

- **Endpoint**: POST /auth/login

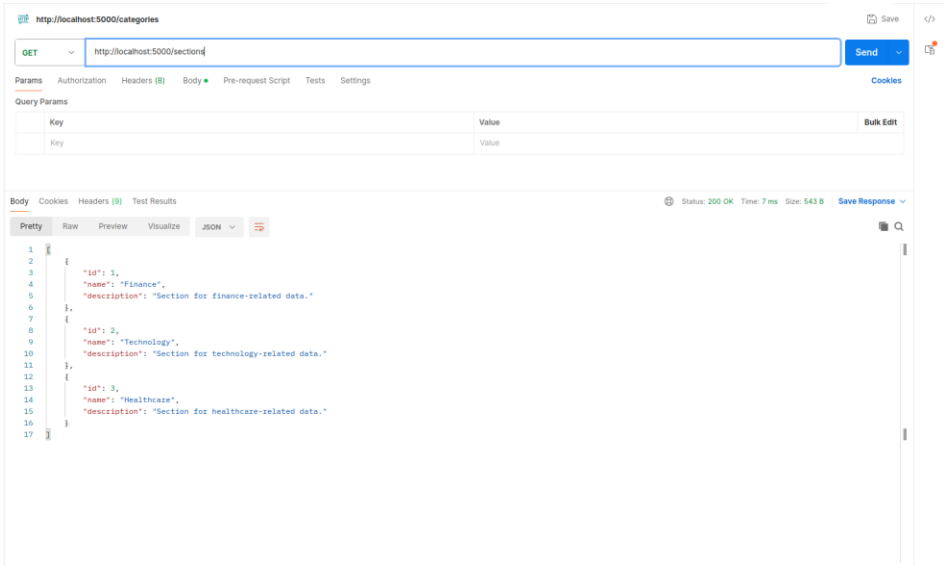


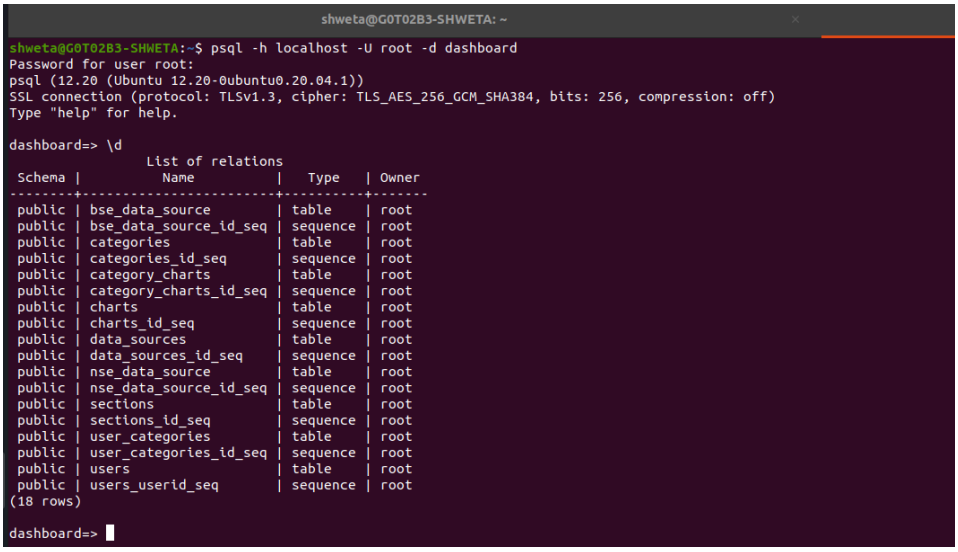
Charts

- **Create Chart Endpoint**: POST /charts
- Create Category-Chart **Endpoint**: POST /charts/category-charts
- **Get All Charts Endpoint**: GET /charts



Sections

- **Create Section Endpoint**: POST /sections
- **Get All Sections Endpoint**: GET /sections
- **Get Section by ID Endpoint**: GET /sections/:id
- **Update Section Endpoint**: PUT /sections/:id
- **Delete Section Endpoint**: DELETE /sections/:id
- 

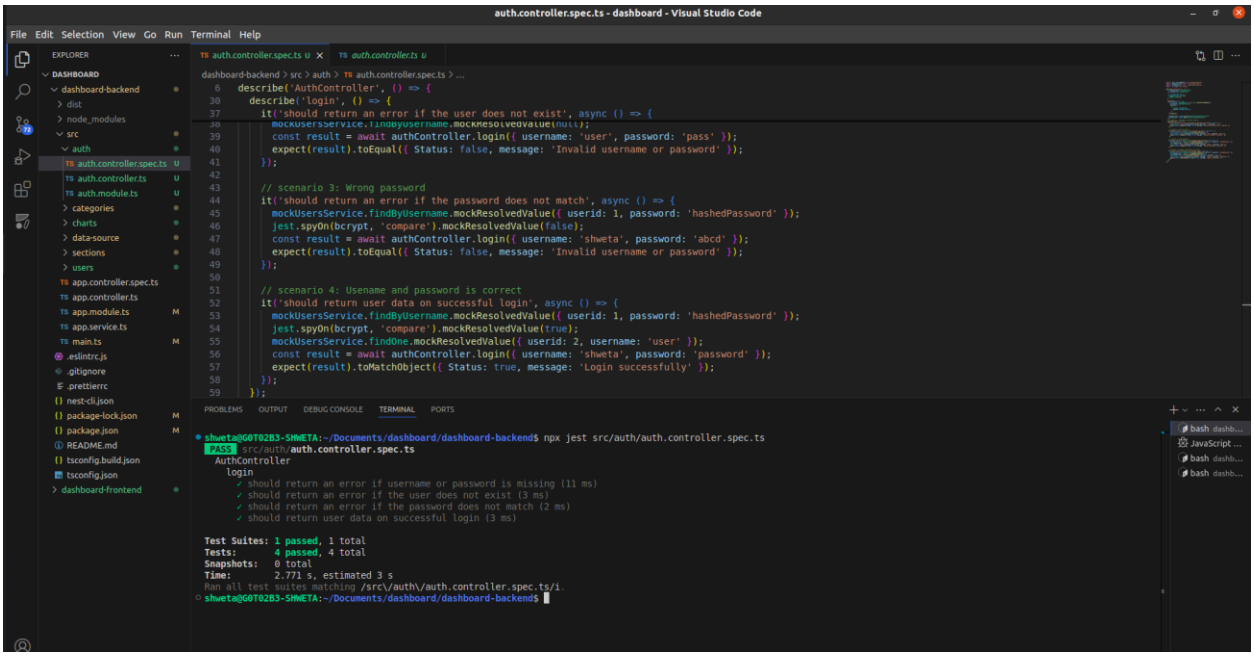


5. Unit Test Cases:

- The tests are implemented using Jest and verify the behavior of the login method.

AuthController:

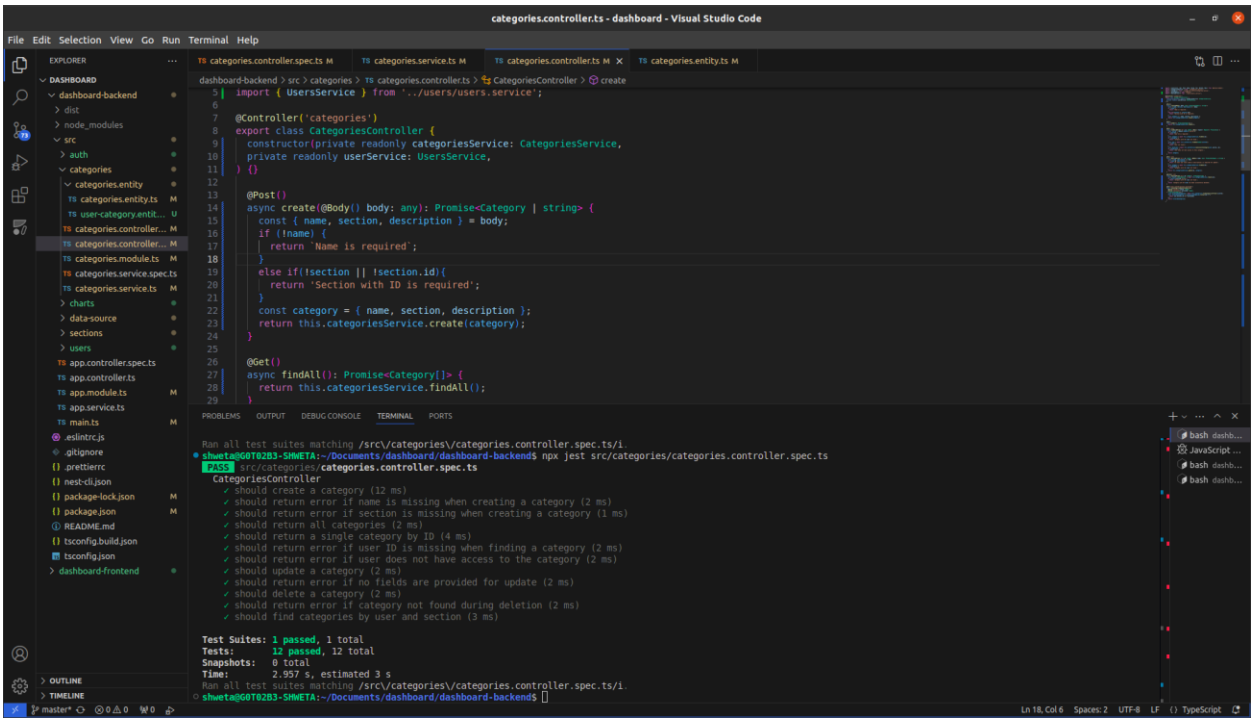
- Scenarios:
    - Username or Password Missing
    - User Does Not Exist
    - Incorrect Password
    - Successful Login



CategoriesController

- Scenarios
    - Create Category Successfully
    - Missing Category Name
    - Missing Section ID
    - Retrieve All Categories
    - Retrieve Category by ID
    - Missing User ID When Retrieving Category
    - No Access to Category
    - Update Category Successfully
    - No Fields Provided for Update
    - Delete Category Successfully

- Category Not Found During Deletion
- Find Categories by User and Section



6. Static Code Quality Report
    a. Report is generated using ESLint

Eslint Report link:
https://github.com/ShwetaPokale/dashboard/blob/master/dashboard-backend/eslint-report.xml